

Environment for distance study of embedded system programming

Margus Ernits
Estonian IT College
Raja 4C
Tallinn, Estonia
Margus.Ernits@itcollege.ee
<http://robot.itcollege.ee>

Kalle Tammemäe
Estonian IT College
Raja 4C
Tallinn, Estonia
Kalle.Tammemae@itcollege.ee
<http://www.itcollege.ee>

Keywords

Distance training, embedded systems, remote control

Abstract

Modern education needs distance learning and e-learning as an alternative to classical approach, and study of embedded systems is not an exception. Current article focuses on specific aspect of distance study of embedded systems. Possible solutions include lendable home-lab kits, virtual labs and distance labs.

Functional and non-functional requirements for usability, deployability and manageability have been developed for experimental distance-lab. Methods and problems in distance study of embedded system using particular distance-lab are briefly described.

To use this lab student needs Internet-connected computer with Java capable web browser. Student receives assignment through web and will prepare a solution using standard editor or IDE, after that C source will be uploaded using dedicated lab system web page. Source code will be compiled remotely and feedback to student will be provided. After successful compilation student can load binary code to embedded system and execute it. User can visually observe all reactions of remote laboratory kit through remote web-camera.

Distance lab is designed using GPL and free software to make it possible to deploy a copy of system at other universities.

This particular lab uses virtualization technologies to simplify manageability and installation process.

Some unfinished areas such as distance debugging, improved feedback and automatic personalization of assignments are described in the paper.

Experiments have proved correctness of the concept and user fidelity of running the remote labs.

This work was supported by EU Structural Funds project: Home and Distance laboratories 1.0101.06-0420

Introduction

Distance study of embedded systems can be implemented in several ways. For embedded system programming student needs access to real hardware and software. Alternative approach is using virtual emulated environment. Some lab works can be done this way but an emulated system does not cover well some specific areas, e.g. learning of interrupt masking technology.

It is possible to lend lab kits to students and let them work with real hardware and software at home [1]. This approach has clear advantages. Student can do hands-on labs with embedded system and in addition will learn how to set up the development environment on their own computer and how to connect remote systems. In principle, student can carry on exactly same work at home as in university laboratory.

Known disadvantage of kits-based lab is their high initial cost and short lifetime. Extra kit and software license is required for every student, whereas usually after only a few years the kits will become outdated. For example, three years old development boards have typically serial or parallel port connectors for programming and debugging proposes. Contemporary laptops do not have old standard serial and parallel communication ports (i.e. RS232, Centronics) and older kits should be equipped with additional USB to serial/parallel converters. Moreover, all home kits will need careful testing and possible service before those can be re-used in following study semester.

Also, student should use several different platforms like 8bit MCUs and 32bit MCU boards to gain wider experience. Lendable ARM9 kits with JTAG dongle are especially expensive.

Our solution is based on concept that programmable systems can be remotely accessed and controlled over Internet. The distance lab does not require special software on student side and simplifies software license management because special software is installed on remote server side only. Only a few kits are required for distance lab once supporting booking system and time-sharing of kits is established.

At the same time, control and observability can be preserved to a large extent in comparison of running the labs locally.

Goals

We have defined the following goals of distance study of embedded system programming:

- Students should be able to run labs any time and in any location;
- All types of embedded systems that are used on real laboratory should be accessible over Internet;
- Students should get direct and informative visual feedback of lab entity;
- No special requirements are defined for student computer (no need for legacy communication interfaces or programming tools), only Internet connection and Java capable web-browser are needed;
- Simple deployability of distance lab system. The server side of the system has to be installable by even moderately skilled system administrator of the institution;
- Easy manageability. System architecture should enable quick restore after hardware or software failure or human error occurrence;
- Readiness to use GPL or GPL-compatible software license policy.

Implementation

Accessing the lab over Internet gives student the advantage of choosing time and place to complete lab assignment. The drawback of this approach is that laboratory equipment can not be touched by hand and students can not connect devices personally. To compensate the lack of real presence, the distance lab uses web cameras to give maximal look and feel as if using laboratory devices at the spot.

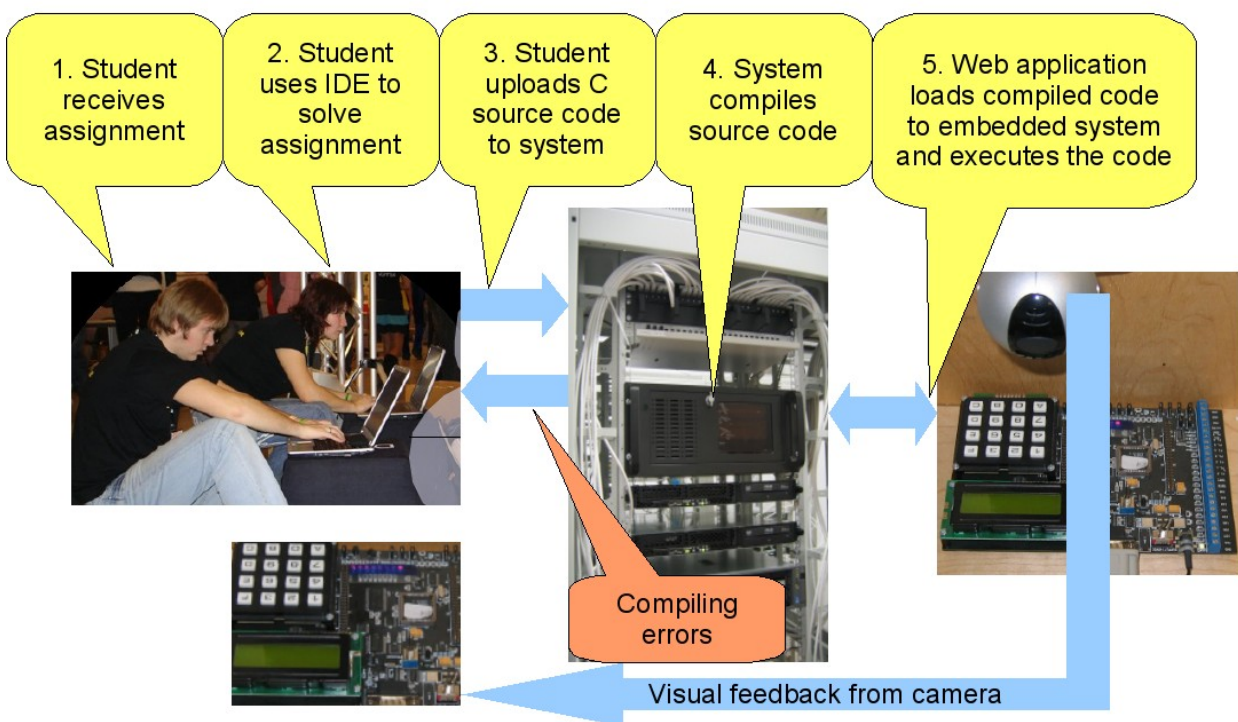


Figure 1: General view of distance laboratory

To use this lab student needs Internet-connected computer with Java capable web browser like Mozilla Firefox, Internet Explorer, Opera or Safari. As shown on Figure 1, the student receives assignment through web and solves it using their own favorite editor or IDE, after that the C source code will be loaded to lab system using dedicated web application. The source code will be compiled by server, and necessary feedback will be returned to student. After successful compilation student can upload binary code to embedded system and execute it. User will get visual feedback of kit's LCD screen and equipped LED's through web-camera.

Architecture

Current version of distance lab has the system architecture as seen on Figure 2. Core of the system contains web application written in PHP5 which runs on Ubuntu Server operating system hosted by VMware Server virtualization product.

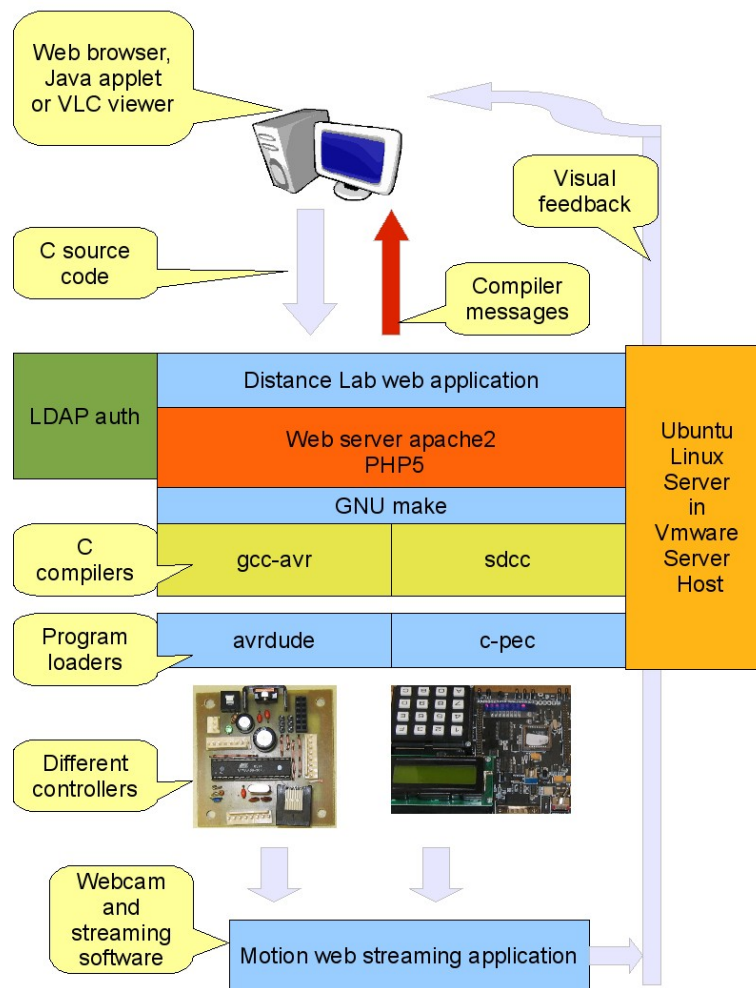


Figure 2: System architecture

Web application has simple file upload form which is used to deploy C source code to server. After uploading, the web application executes *GNU make* with controller dependent makefile. Each controller type has slightly different makefile to compile and upload binary code to embedded system. Web interface is same for different controller types. Adding a new controller type is relatively simple and contains following tasks: makefile rewrite, compiler installation, loader installation and file copy. Minimum hardware required for running one lab consists of PIII800MHz

computer, equipped with 128MB of ram and USB1.1 web camera. Of course, more and better resources will improve the feedback significantly. The most resources consuming task is picture streaming and delivering from web camera.

Virtualization

To ensure fast deployment cycle and seamless manageability, virtualization technologies can be used. Distance laboratory is deployed as a virtual machine image, where all lab software is pre-installed and configured.

Several virtualization technologies exist and are free or even GPL'ed products widely used in production environments. Distance labs have some specific requirements on virtualization technologies:

1. Virtualization must support USB devices like web cameras. (this requirement excludes most of mature products because of lack of proper USB support[2]).
2. Virtualization software must support manually configured redirection of serial and parallel interfaces from host to guest. Every lab should get direct access to their programming and debugging dongles after booting. It is not wise to connect them manually after reboot of host machine.
3. Virtualization software should be able to run on an older computer without Intel VT or AMD-V extensions equipped processor.
4. Host computer should support Linux and Windows operating systems.

Those requirements are currently met with VMware Server virtualization product which is used in distance laboratory.

Manageability and deployability

Using virtualization technologies gives the advantage of restoring image of virtual machine to any VMware equipped host after server hardware failure or upgrade. The only restriction is availability of usable communication interfaces. This feature also provides a simple way to deploy new installation to any host computer.

Distance lab is often connected to untrusted environment and needs regular and automatic upgrading of guest operating system and applications. This is useful for security reasons but sometimes software failsto run after upgrading. In such case the system administrator can easily go back to previous save point of virtual machine and restore working state to get more time to investigate reasons of failure of the upgraded system.

Web services and management secure shell service can be monitored using service and network monitoring software Nagios 3 [5].

Security considerations

Connecting laboratory to Internet demands more attention to security. It is good to have open lab for demonstration purposes but in real environment reliable authentication and authorization means are a must. In our case, production environment authenticates and authorizes against institution LDAP using standard PHP LDAP and apache2 LDAP modules [3].

For increased reliability, unneeded software is removed from distance lab virtual image preventing unapproved usage of software weaknesses and keeping the core small and fast.

Experiments

The environment of distance study has been piloted in 2007 in course Hardware close programming. Initial feedback from users was positive. But students mentioned they expected more interactivity from lab. Adding some actuators like servo motors can solve this issue. Teachers proposed that user should be able to see states of registers to get more advanced feedback from system. Lack of possibility to book distance lab sessions was also a problem on hectic times before examinations.

Working demonstration of distance lab accessible over Internet [6]

Future work

Study of embedded system programming is permanent part of CE curricula and although basics of study are stable, application areas are changing fast. Students need more accurate knowledge about testing and debugging of different embedded systems. The distance lab environment should be able to adapt to changing educational needs. Growing user base demands implementing time booking for the lab sessions. Teachers need functionality interfaced with electronic study environment to generate personalized homework assignments for each student. Semi-automatic grading of homework is also desired to optimize teachers work.

Future development areas:

- Debugging of embedded system using JTAG with GNU debugger GDB,
- Lab session booking system,
- Semi-automatic generation of personalized homework assignments,
- Semiautomatic grading system.

Most complicated areas here are debugging and generation of personalized homework. Idea of new architecture is shown on Figure 3.

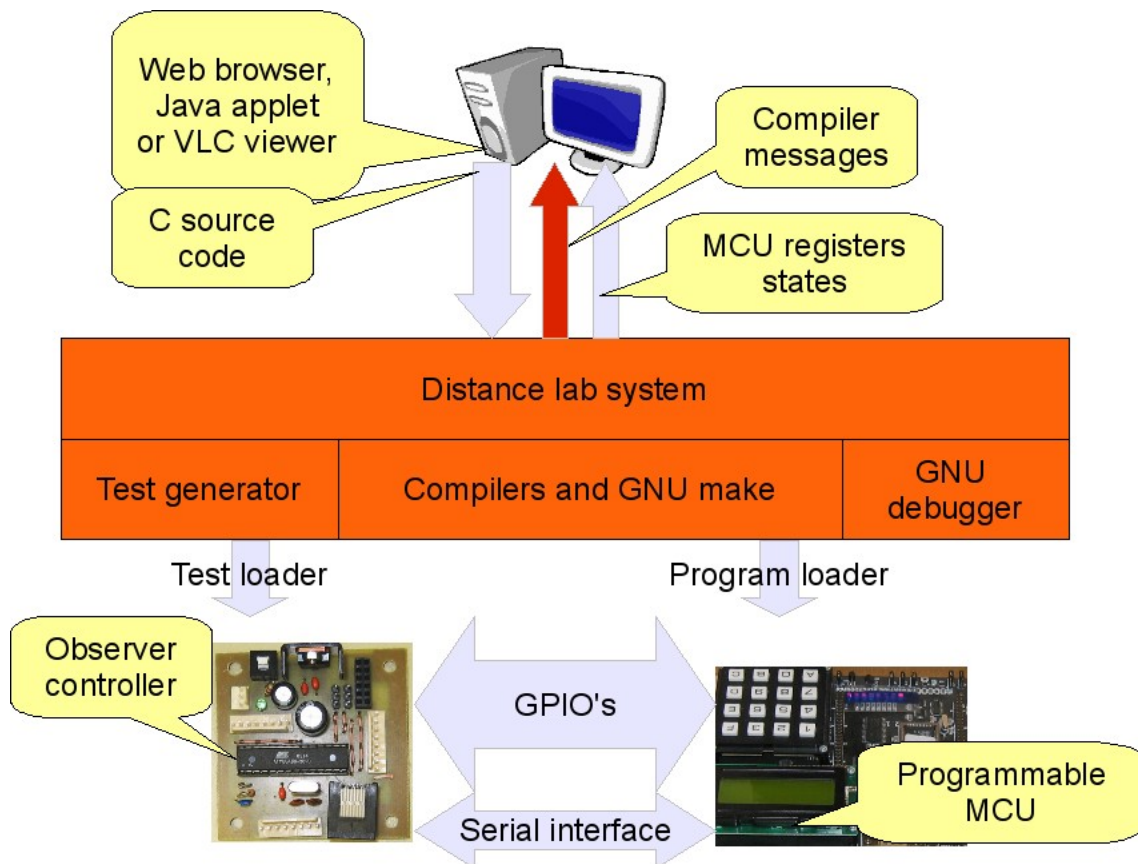


Figure 3: Future architecture

Register level debugging can be implemented using GNU debugger and JTAG interface. Students should be able to see content of MCU registers.

Personalization can be done using student's ID code which is used to generate simple communication protocol. Protocol description will be displayed to student who should implement this protocol in MCU. Automated testing of student protocol can be done by observer controller.

Conclusion

Distance study of embedded systems can be implemented using web technologies. Manageability and deployability issues can be handled using contemporary virtualization technologies. The feedback from the pilot deployment semesters has been analyzed and the areas of future development have been defined.

This distance laboratory is currently in use at Estonian IT College for teaching hardware close programming.

References

1. "Hands-On Lab Exercises", Mai 2008Hrsg: Interstudy Project ISBN: 978-9955-20-340-7
2. Wikipedia Comparison of virtual machines
http://en.wikipedia.org/wiki/Comparison_of_virtual_machines
3. Apache Module mod_auth_ldap http://httpd.apache.org/docs/2.0/mod/mod_auth_ldap.html
4. LDAP in PHP <http://ee.php.net/manual/en/book.ldap.php>
5. Monitoring software Nagios <http://www.nagios.org/>
6. DEMO lab <http://robot.itcollege.ee:800/~robot/51Kit0>